Multiple Class Image-based Vehicle Classification using Soft Computing Algorithms

Sanjivani Shantaiya[1], Kesari Verma[2], and Kamal Mehta[3]
[1]Department of Computer Science and Engineering, CSV Techical University, India
[2]Department of Computer Application, National Institute of Technology, India
[3]Department of Computer Science and Engineering, Nirma Univesity, India

Abstract: *Automatic vehicle classification has expanded into a momentous topic of study due to its importance in autonomous navigation, traffic analysis, surveillance security systems, and transportation management. This paper presents multiclass image based classification of vehicles like bikes and cars using soft computing algorithms like artificial neural network decision tree and support vector machine as well. The objective of this paper is to automate the classification of vehicles from images. A good set of descriptor features that capture the most important properties of an object are used to identify the object uniquely. Different views of vehicle images are considered as important factor during classification process as six multiple class labels are created accordingly.*

Keywords: *Detection, classification, artificial neural network, decision tree, support vector machine.*

Received April 25, 2014; accepted June 7, 2015

1. Introduction

The field of automatic traffic monitoring systems is today of huge interest as a result of to its implications within the prospects of security. Surveillance of vehicular traffic offers a context for the extraction of significant information like scene motion and traffic statistics, object classification, vehicle identification, anomaly detection, moreover because the analysis of interactions between vehicles. A vehicle classification system is crucial for effective transportation systems (e.g., traffic management and toll systems), parking, optimization, enforcement, autonomous navigation, etc.

2. Motivation

With economic development the number of various traffic vehicles has grown speedily. Therefore, the demands on the traffic safety surveillance system have dramatically increased. According to government statistics, traffic violation is the main cause of traffic accidents. To reduce traffic violation situations, human based manual monitoring is not possible because of limited police man-power. Problems with the existing monitoring systems are:

- Human operators are required to monitor activities.
- Require a lot of human intervention to locate the same object in case the multiple cameras are used.
- Major research is based on classification of two objects only.

Since the existing system is not efficient and exact; this motivates to set up a system which is based on classification of vehicles images. This research work classifies the vehicles automatically from image considering different views of vehicles accordingly into multiple classes.

3. Related Work

Various approaches to vehicle classification and detection are noted and mentioned in the field of computer vision. Wei *et al.* [11] use a 3-D parameterised model which corresponds to features of the vehicles topological structure, classified using a neural network. Jun Wei Hsieh *et al.* [5] proposes a novel traffic surveillance system for detecting, tracking, and recognizing vehicles from different video sequences. This system includes an optimal classifier which is designed to robustly categorize vehicles into different classes. Gupte *et al.* [4] presents an algorithm for detection and classification of vehicles in image sequences of traffic scenes. The system classifies vehicles into two categories cars and non-cars (e.g., buses, trucks, SUVs). Kato *et al.* [8] propose the development of a driver assistance system using a vision based preceding (vehicles travelling in the same direction as the subject vehicle) vehicle recognition method, which is capable of recognising a wide selection of vehicle types against road environment backgrounds. The system classifies vehicles into three different categories. Dubuisson Jolly *et al.* [7] uses a deformable template algorithm consisting of finding a template that best characterises the vehicle into one of five categories. Similarly, Margrit Betke *et al.* [2] developed a real-time vision system that analyzes colour videos taken from a forward-looking video camera in a car driving on a highway. The system uses a combination of colour, edge, and motion information to recognize and track the road boundaries, lane markings and other vehicles on the road. Zhong Qin *et al*. [11] propose the method of vehicle classification using image analyses. They set up road background according to the serial images, and the vehicle region are segmented using background divide, then its moment invariant features are calculated which were provided as input to Back Propogation (BP) neural networks with three layers, and the vehicle type is classified according to the output of the BP neural networks.

Despite the large amount of research in vehicle detection, there has been comparatively very little focused the field of vehicle classification. Most systems either detect (locate a vehicle against a background) or classify vehicles into broad categories such as cars, buses and trucks [4, 7, 8, 9, 11, 12, 14, 15]. It's been observed that very little amount of work has been done on classification of multiple objects. This research paper focuses on classifying multiple objects. Although a number of vehicle classification system exists most of them seek only to distinguish between categories of heavy vehicles only but the proposed research concentrates both on heavy vehicles as well as light vehicles. Most significantly it distinguishes vehicles according to its view or orientation. This paper uses three classification algorithms and also discuss about their performance analysis.

4. Proposed Methodology

The task of vehicle classification is shown in given Figure 1 which describes proposed solution.

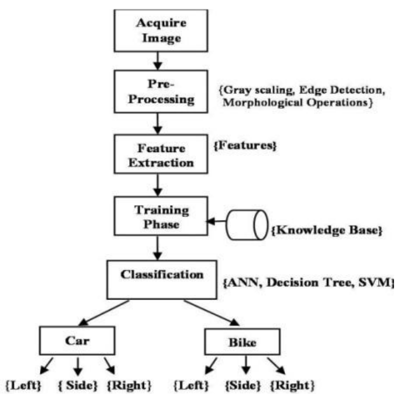


Figure 1. Proposed methodology.

The image based vehicle classification starts with inputting the acquired image and thereafter pre-processing it. Pre-processing includes grayscale conversion, edge detection and morphological operations. After the pre-processing next phase is extracting the morphological features that capture the most important properties of an object to create knowledge base. This knowledge base is used to train the system by applying soft computing algorithms like ANN, SVM and decision tree. Finally, in the classification phase the unknown test images are classified into six multiple classes.

5. Image Pre-processing and Morphology

The algorithm for pre-processing and morphological operations is as given below.

Algorithm 1: Pre-processing and morphological operations.

Input: *Vehicle and pedestrian database* $VT=\{VP_1; VP_2; VP_3; ...; VP_n\}$ *where vehicle parameters* $V_i=(a_{i1}, p_{i1}, ma_{i1}, mi_{i1}, c_{i1}, ec_{i1}, eq_{i1}, s_{i1}, ..., (a_{in}, p_{in}, ma_{in}, mi_{in}, c_{in}, ec_{in}, eq_{in}, s_{in})$
Output: *Pre-processed Images.*

Start

1. *Extract image from the dataset. for* $i=1$: V_n

Do

2. *Convert RGB image* I_i *to Grayscale image* I_g *as*: $Ig=rgb2gray(I_i)$.
3. *Apply function edgeDetection*().

{

4. *Compute* f_x *and* f_y.
5. $fx = \partial \frac{\partial}{\partial x}(f * G) = f * \frac{\partial}{\partial x} G = f * G_x$.
6. $fy = \partial \frac{\partial}{\partial y}(f * G) = f * \frac{\partial}{\partial y} G = f * G_y$.
   - *G*(*x*, *y*) *is Gaussian function.*
   - *G*(*x*, *y*) *is derivative of G*(*x*, *y*) *with respect to x*: $G(x, y)=\frac{-x^2}{\sigma^2} G(x, y)$.
   - *G*(*x*, *y*) *is derivative of G*(*x*, *y*) *with respect to y*: $G(x, y)=\frac{-y^2}{\sigma^2} G(x, y)$.
7. *Compute the gradient magnitude.*

$$magn(i,j) = \sqrt{{f_x}^2 + {f_y}^2}$$

8. *Apply non-maxima suppression.*
9. *Apply hysteresis thresholding/edge linking.*

}

10. *Apply morphological function based on structuring element.*
11. *Extract features of object as*: $\{a_i, p_i, ma_i, mi_i, ci, ec_i, eq_i, s_i\}$ $=\{$*bw.area*(), *bw.perimeter*(), *bw.MajorAxis*(), *bw.MinorAxis*(), *bw.ConvexArea*(), *bw.Eccentricity*(), *bw.Equivdiameter*(), *bw.Solidity*()$\}$.

End

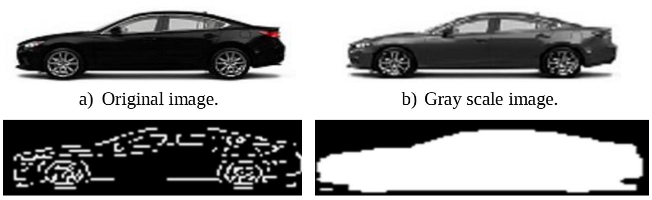
The scope and complexity of the vehicle classification problem considered in this paper is exemplified by the database of cars and bikes images, illustrated in Figures 2 and 3, with sample images of different models and versions of the cars and bikes. Images are downloaded from various websites of cars and bikes of different companies. In the experiments, 300 images of cars and bikes from different views (each of side view, left side and right side view of cars and bikes) are used for registration (training) and out of which 60 (each of side view, left side and right side view of cars and bikes) for evaluation (testing) for the proposed system. After acquiring image from database it is pre-processed for further enhancement. During the pre-processing phase acquired image first gets converted into gray scale image. Then, image is been resized in proper dimension. After then edges of gray scale image is been find out using canny edge detector. Later on morphological operations has been performed for further processing.

Figure 2. Sample images of cars with different views.

Figure 3. Sample images of bikes with different views.

Morphological image processing refers to a category of algorithms that transforms the geometric structure of an image. Morphology can be used on binary and gray scale images, and is helpful in several areas of image processing, like skeletonization, edge detection, restoration and texture analysis. The fundamental mathematical morphology operations dilation, erosion and closing are used on image [4].

All the above morphological operations are applied on image which is shown in Figure 4.



a) Original image. b) Gray scale image.

c) Edge detected figure. d) Image after morphological operations.

Figure 4. Morphological operations.

6. Feature Extraction

After pre-processing of image features has to be extracted. A good set of descriptor features should include the features that capture the most important properties of an object and can be used to identify the object uniquely. Sample dataset having features is shown in Table 1. An object can be identified by its two or three dimensional geometrical properties. Such properties could be described as:

- *Area*: Actual number of pixels within the object region. The total number of pixel covered by the object, area is calculated using method known as region props that measures a set of properties for each connected component (object) within the binary image, BW. The image BW is a logical array; it can have any dimension.
- *Perimeter*: The distance around the boundary of the region. Perimeter is computed by calculating the distance between each adjoining pair of pixels around the border of the region [6]. The perimeter of an object is given by the integral as follows:

$$T = \int \sqrt{2^{x} + 2^{y}}\,dt \tag{1}$$

This perimeter is used for the parametric boundary representation. By the aid of a boundary following algorithm, the object perimeter can be found out. If 1_x, ..., n_x is a boundary coordinate list, the object perimeter is given by:

$$T = \sum_{i=1}^{N-1} d_i = \sum_{i=1}^{N-1} \left| x_i - x_{i+1} \right| \tag{2}$$

Major Axis: - Major and minor axes are the simplest of all features but yet important. They give essential information of an object such as elongation, eccentricity etc. The major axis points are the two points in an object where the object is more elongated and where the straight line drawn between these two points is the longest. Major axis points are calculated by all possible combinations of perimeter pixels where the line is the longest [6]. The length of the major axis is given by:

$$MajorAxisLength = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

Where (x1, y1) and (x2, y2) are the coordinates of the two end points of the major axis.

- *Minor Axis*: The minor axis is drawn perpendicular to the major axis where this line has the maximum length. Once the end points of the minor axis have been found, its length is given by the same equation as the major axis length. It is also called the object width [6].
- *Convex Area*: It specifies the number of pixels in 'Convex Image' [6].
- *Eccentricity*: Eccentricity is the ratio between the lengths of the short axis to the long axis Gonzalez *et al.* (2004) [3] as defined in the following equation:

$$Eccentricity = \frac{Axislength_{short}}{Axislength_{long}} \tag{4}$$

The value of *Eccentricity* is between 0 and 1. Eccentricity is also called ellipticity with respect to minor axis and major axis of the ellipse. If the major axis gets longer, eccentricity gets higher.

- *Equivdiameter*: Specifies the diameter of a circle with the same area as the region [6]. It is computed as:

$$Equidiameter = \sqrt{\frac{4 * Area}{pi}} \tag{5}$$

- *Solidity*: In simple terms density is mass per unit volume. But in two dimensional image objects this can be defined as the ratio between the area and convex area of the same object [6]:

$$Solidity = \frac{Area}{ConvexArea} \tag{6}$$

Table 1. Sample parameters.

Area	Perimeter	Major Axis	Minor Axis	Convex Area	Eccentricity	Equiv Diameter	Solidity
822	257.137	68.6954	24.5773	1592	0.933809	32.3512	0.516332
464	150.953	46.8993	20.4605	722	0.899818	24.306	0.642659
556	191.439	57.6385	21.5288	1098	0.927624	26.6068	0.506375
782	205.539	61.2711	24.8894	1249	0.913777	31.5543	0.626101
757	218.024	67.87	24.5518	1286	0.932276	31.0458	0.588647

7. Classification Using ANN

Algorithm for vehicle classification using ANN is as shown in Algorithm 2. To classify vehicles multilayer feed forward neural network is employed. A multilayer feed forward neural network is an interconnection of perceptrons during which data and calculations flow in a single direction, from the input data to the outputs. The number of layers in a neural network is the number of layers of perceptrons [1]. In order to train the neural network, a set of training vehicle images were needed, and the images were predefined.

Algorithm 2: Vehicle Classification Using ANN

Input: *Vehicle Parameters* $V_i = \{(a_{i1}, p_{i1}, ma_{i1}, mi_{i1}, c_{i1}, ec_{i1}, eq_{i1}, s_{i1}) \ldots, (a_{in}, p_{in}, ma_{in}, mi_{in}, c_{in}, ec_{in}, eq_{in}, s_{in})\}$.
Output: *Multiple classes of vehicles.*

Start

1. *Input image parameters from the dataset to feed forward neural network.*
$\{(a_{i1}, p_{i1}, ma_{i1}, mi_{i1}, c_{i1}, ec_{i1}, eq_{i1}, s_{i1}) \ldots, (a_{in}, p_{in}, ma_{in}, mi_{in}, c_{in}, ec_{in}, eq_{in}, s_{in})\}$.
2. *Take hidden layers as* 10.
3. *Calculate weighted sum of inputs for hidden node as*:

$$zh = \sum_{j=0}^{d} w_{hj} x_j$$

4. *Compute sigmoid function applied at the hidden node as*:

$$sigmoid(a) = \frac{1}{1 + e^{-a}}$$

5. *Compute the output for multiple classes as*:

$$y_i = \frac{e^{oi}}{\sum_{i=1}^{K} e^{oi}}$$

6. *Error is calculated and back propagated to change the weight which helps to improve accuracy*.

$$E = \frac{1}{2}\sum_{i=1}^{K}(t_i - y_i)^2$$

Where t_i *is the target output, and* y_i *is the actual output of the network*

End

The artificial neural network is used to train system which uses patternnet as a network which is shown in Figure 5. ANNs based patternnet are feed forward networks that can be trained to classify inputs according to target classes. The training method trainlm was used to train these networks. The hidden layer contains 10 neurons. During training, the connection weights of the neural network were initialized with some random values. The training samples within the training set were input to the neural network classifier in random order and the connection weights were adjusted according to the error back-propagation learning rule. This process was recurrent till the Mean Squares Error (MSE) fell below a predefined tolerance level or the maximum number of iterations is achieved. Once the network training was finished, the network was tested with test dataset (60 images), and therefore the classification accuracy was calculated. The performance goal was met.

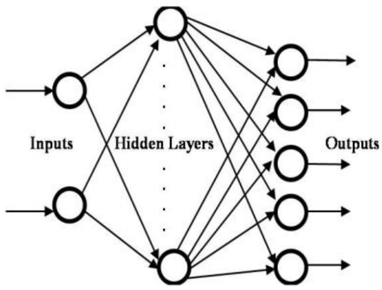


Figure 5. Structure of ANN.

8. Classification Using Decision Tree

Decision tree is stratified like tree structure, wherever every internal node denotes a test on an attribute, every branch denotes an outcome of test, and every leaf node holds a class label. Let the training dataset be T with class labels $\{C_1, C_2, \ldots, C_i\}$.The decision is built by repeatedly partitioning the training data till all the records in partition belong to some class [1]. The uppermost node in a tree is the root node. Given a tuple, X, for which the associated class label is unknown, attribute values of the tuple are tested against decision tree. A path is traced from the root to a leaf node that holds the class prediction for that tuple. Figure 6 shows the structure of the tree that was generated after training showing six different classes.

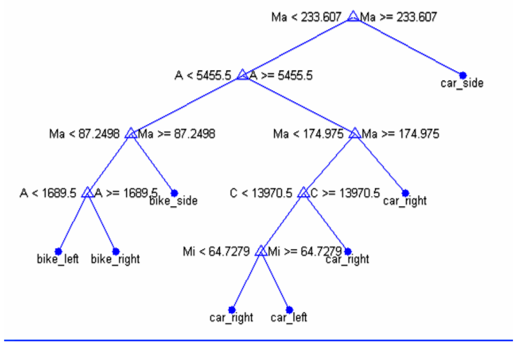


Figure 6. Decision tree.

9. Classification using Multiclass SVM

A Support Vector Machine (SVM) is a discriminative classifier defined by a separating hyperplane. The main idea behind this classification technique is to Separate the classes with a surface that maximise the margin between them, using boundary pixels to create the decision surface. The data points that are closest to the hyperplane are termed "support vectors". SVM were initially designed for binary (two class) problems. When dealing with multiple classes, an appropriate multi-class method is needed. Vapnik [13] suggested comparing one class with the others taken together. This strategy generates n classifiers, where n is the number of classes. The final output is the class that corresponds to the SVM with the largest margin, as defined above. For multi-class problems one has to determine n hyperplanes. Thus, this method requires the solution of n Quadratic Programming (QP) optimization problems, each of which separates one class from the remaining classes. This strategy can be described as 'one against the rest'.

Consider an M-class problem, where we have N training samples: {x1, y1}, …,{xN, yN} Here, xi ε Rm is a m-dimensional feature vector and yi ε {1, 2, …, M} is the corresponding class label. One-against-all approach constructs M binary SVM classifiers, each of which separates one class from all the rest. The i^{th} SVM is trained with all the training examples of the ith class with positive labels, and all the others with negative labels [10]. Mathematically, the i^{th} SVM solves the following problem that yields the i^{th} decision function

$$f_i(x) = w_i^T \varphi(x) + b_i \tag{7}$$

minimize:

$$L\left(W, \xi_j^i\right) = \frac{1}{2} \parallel w_i \parallel^2 + C \sum_{l=1}^{N} \xi_j^i \tag{8}$$

subject to:

$$\tilde{y}_j\left(w_i^T \phi\left(x_j\right) + b_i\right) \geq 1 - \xi_j^i, \xi_j^i \geq 0 \tag{9}$$

Where $\tilde{y}_j = 1$ if $y_j = i$ and $\tilde{y}_j = -1$ otherwise.

At the classification phase, a sample x is classified as in class i* whose fi* produces the largest value.

$$i^* = \arg\max f_i(x) = \arg\max(W_i^T \phi(x) + b_i) \tag{10}$$

where i=1, …, M.

10. Comparing Classifiers

Evaluation of the performance of classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as a confusion matrix as shown in Table 2, 3 and 4. Although confusion matrix provides the information needed to determine how well a classification model performs, summarizing this information would make it more convenient to compare the performance of different models. This can be done using performance metric such as accuracy which is defined as:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{11}$$

Equivalently, the performance of model can be expressed in terms of it error rate, which is defined as:

$$Error\ rate = \frac{Number\ of\ wrong\ predictions}{Total\ number\ of\ predictions} \tag{12}$$

Classification algorithms seek models which should attain highest accuracy and eventually lowest error rate when applied to test data set.

Table 2. Confusion matrix using ANN.

			Predicted					
			Car			**Bike**		
			Side	**Right**	**Left**	**Side**	**Right**	**Left**
Actual	**Car**	**Side**	8	1	1	0	0	0
		Right	1	8	1	0	0	0
		Left	0	1	7	2	0	0
	Bike	**Side**	0	0	1	8	1	0
		Right	0	0	1	1	7	1
		Left	0	0	0	1	1	8

Table 3. Confusion matrix using decision tree.

			Predicted					
			Car			**Bike**		
			Side	**Right**	**Left**	**Side**	**Right**	**Left**
Actual	**Car**	**Side**	8	1	1	0	0	0
		Right	1	8	1		0	0
		Left	0	1	9	0	0	0
	Bike	**Side**	0	0	1	8	1	0
		Right	0	0	0	1	8	1
		Left	0	0	0	1	1	8

Table 4. Confusion matrix using multiclass SVM.

			Predicted					
			Car			**Bike**		
			Side	**Right**	**Left**	**Side**	**Right**	**Left**
Actual	**Car**	**Side**	9	1	0	0	0	0
		Right	1	9	1	0	0	0
		Left	0	0	10	0	0	0
	Bike	**Side**	0	0	1	8	1	0
		Right	0	0	0	0	9	1
		Left	0	0	0	0	1	9

11. Conclusions

Vehicle recognition is important technology for road traffic monitoring, management and security issues. However, it is difficult task for computer to attain because vehicles have a wide range of different appearances due to variety of their shape, colours and orientation specifically. This paper proposes a novel vehicle recognition process based on vehicle type as well as its orientation or view. It identifies vehicles as cars or bikes from left, right and side view. Acquired images were first preprocessed; then features that capture the most important properties of an object to identify the object uniquely were extracted. A pattern recognition neural network is used to classify the vehicle images. In the test dataset, the classification accuracy is 75.00 %. Also, decision tree is used to classify the dataset against given attributes. Classification accuracy of decision tree is 82.00%. Finally, multiclass SVM classifier is applied on the dataset which achieves highest accuracy i.e., 90% as shown in Figure 7. Although, neural network is simple and easy to maintain but there are many free parameters, such as, the number of hidden nodes, the learning rate, minimal error, which may greatly influence the final result. Decision tree perform classification without much computation and can generate understandable rules but it may perform poorly with many class and small data. Also, the process of growing decision tree is computationally expensive. The major strengths of SVM are the training is relatively easy. No local optimal, unlike in neural networks. It scales relatively well to high dimensional data and the trade-off between classifier complexity and error can be controlled explicitly provided good kernel function is selected. SVMs act as one of the best approach as it outperforms amongst the three methods.

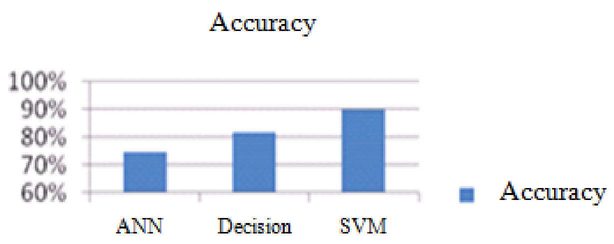


Figure 7. Performance comparison graph of classifiers.

12. Future Scope

The present work can be extended for video processing where input will be a video having vehicles and then classification will be done on video based datasets. Additionally alternative other descriptors like boundaries and sub segments may also be extracted to extend accuracy. Present work can also extended by applying some other classifiers like Adaboost, PSO etc., and then compare their performance as well.

References

[1] Alpaydin E., *Introduction to Machine Learning*, MIT Press, 2009.

[2] Betke M., Haritaoglu E., and Davis L., "Real-Time Multiple Vehicle Detection and Tracking from a Moving Vehicle," *Machine Vision and Applications*, vol. 12, no. 2, pp. 69-83, 2000.

[3] Gonzalez R., Woods R., and Eddins S., *Digital Image Processing using Matlab*, Gatesmark Publishing, 2009.

[4] Gupte S., Masoud O., Martin R., and Papanikolopoulos N., "Detection and Classification of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37-47, 2002.

[5] Hsieh J., Yu S., Chen Y., and Hu W., "Automatic Traffic Surveillance System for Vehicle Tracking and Classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 175-187, 2006.

[6] Huque A., *Shape Analysis and Measurement for the HeLa Cell Classification of Cultured Cells in High Throughput Screening*, School of Humanities and Informatics, 2006.

[7] Jolly M., Lakshmanan S., and Jain A., "Vehicle Segmentation and Classification using Deformable Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 293-308, 1996.

[8] Kato T., Ninomiya Y., and Masaki I., "Preceding Vehicle Recognition based on Learning from Sample Images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 252-260, 2002.

[9] Lipton A., Fujiyoshi H., and Patil R., "Moving Target Classification and Tracking from Real-Time Video," *in Proceeding of the 4th IEEE Workshop on Applications of Computer Vision*, New Jersey, pp. 8-14, 1998.

[10] Liu Y. and Zheng Y., "One-Against-All Multi-Class SVM Classification Using Reliability Measures," *in Proceeding of IEEE International Joint Conference on Neural Networks*, Montreal, pp. 849-854, 2005.

[11] Qin Z., "Method of Vehicle Classification based on Video," *in Proceeding of the IEEE /ASME International Conference on Advanced Intelligent Mechatronics*, China, pp. 162-164, 2008.

[12] Rehman A. and Saba T., "An Intelligent Model for Visual Scene Analysis and Compression," *The International Arab Journal of Information Technology*, vol. 10, no. 2, pp. 126-136, 2013.

[13] Vapnik V., *The Nature of Statistical Learning Theory*, Springer, 1999.

[14] Wu W., QiSen Z., and Mingjun W., "A method of Vehicle Classification using Models and Neural Networks," *in Proceeding of Vehicular Technology Conference* Spring, Greece, pp. 3022-3026, 2001.

[15] Yoshida T., Mohottala S., Kagesawa M., and Ikeuchi K., "Vehicle Classification Systems with Local-Feature Based Algorithm using CG Model Images," *IEICE Transactions on Information and Systems*, vol. 11, no. 12, pp. 1745-1752, 2002.

Sanjivani Shantaiya completed B.E (CSE) from Amravati university, Maharashtra, India and M.tech from technical university of Chhattisgargh, Bhilai, C.G,India. Pursuing Ph.D in CSE from C.V. Raman University, Bilaspur, C.G., India.

Kesari Verma Post graduated in 1998 and got doctorate in 2007. She is current Assistant professor in National Institute of Technology, Raipur, C.G, India She has 12 year R and D and Teaching experience. She has more than 30 publications in journals and conferences.

Kamal Mehta obtained his Bachelor of Engineering in Computer Technology from KITS Ramtek Nagpur University and M. Tech from GEC Raipur Chhattisgarh. He received Doctoral Degree from Technical University of Chhattisgarh State. He is Associate professor at Nirma University,Computer Science and Engineering department,Ahmedabad, Gujrat, India. He has over 16 years of teaching experience. He is credited with 17 publications in reputed journals and 32 papers in conference proceedings. He was principal investigator of AICTE sponsored research project for two years. He is active member of IEI, ISTE, CSI and VLSI society of India.